Designing and Deploying an Information Awareness Interface

JJ Cadiz, Gina Venolia, Gavin Jancke, Anoop Gupta

August 20th, 2002

Technical Report
MSR-TR-2002-87

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Designing and Deploying an Information Awareness Interface

JJ Cadiz, Gina Venolia, Gavin Jancke, Anoop Gupta
Collaboration & Multimedia Systems Team
Microsoft Research
One Microsoft Way, Redmond, WA 98052
{jjcadiz; ginav; gavinj; anoop} @microsoft.com

ABSTRACT
The concept of awareness has received increasing attention over the past several CSCW conferences. Although many awareness interfaces have been designed and studied, most have been limited deployments of research prototypes. In this paper we describe Sideshow, a peripheral awareness interface that was rapidly adopted by thousands of people in our company. Sideshow provides regularly updated peripheral awareness of a broad range of information from virtually any accessible web site or database. We discuss Sideshow's design and the experience of refining and redesigning the interface based on feedback from a rapidly expanding user community.

Keywords
Situational awareness, peripheral awareness, awareness, computer mediated communication, information overload

1 INTRODUCTION

Interfaces that help people stay aware of information without being overwhelmed or distracted are of key concern as more of the people we interact with, information we are interested in, and objects we work with are available digitally. The word "awareness" did not appear in a paper title in the first five CSCW conferences, but appeared eight times in the next five and eleven times in the most recent five conferences. Papers focusing on awareness and notification have increased from one or two in early conferences to a quarter of all papers more recently.

Unfortunately, this literature contains few unequivocal success stories: few systems have been widely deployed. Many (including one created by an author of this paper) have not succeeded in convincing people to use them outside of lab studies or deployments within a research group. Numerous factors can contribute to limited deployment of prototypes: research teams may be small and lack the resources to build deployable software, researchers may not have access to a receptive audience, and prototypes may be designed to test a particular hypothesis and not to appeal to a broader audience. Although the ideas and lessons generated by such prototypes are valuable, they leave a critical question: Why did these prototypes fail to provide users with substantial value relative to cost? What combination of features, design, and process will help an application succeed in establishing a healthy user population?

Sideshow started as one more idea for an interface designed to provide users with peripheral awareness of important information. Rather than concentrate on a specific awareness issue, the research team set out to incorporate a range of features into a versatile and extensible system for dynamic information awareness that could be easily deployed, extended by third parties, and quickly evolved in response to users' experiences.

What happened was something akin to an epidemic within our company. Within a year of being first made available, 13,000 employees installed Sideshow and 7,000 were using it on a regular basis. From logs of system use and interviews of selected users we identified a range of factors contributing to this success.

First, the experience confirms that there is a tremendous wealth of data available on the Internet and corporate intranet that people would like to maintain in peripheral awareness and access with minimal effort. Second, users are willing to give up a portion of their screen space permanently for a peripheral awareness application if the information presented is easily customized and individually relevant. This requires an extensible architecture and a toolkit that allows the community to easily add new sources of information. Third, using the peripheral interface as a launch point for easy drill down substantially increased the amount and value of information made accessible to the users.

In this paper we discuss Sideshow's design and deployment experience. We present the principles that informed the initial design, the iterative design process used during its deployment, and refinements made to balance a broad and sometimes conflicting set of user desires. We present data on what users found valuable and discuss an ecosystem that allowed the user community to continually add value.

The next section discusses prior awareness research that helped inform the design of Sideshow. Section 3 presents

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CSCW'02, November 16–20, 2002, New Orleans, Louisiana, USA.
Copyright 2002 ACM 1-58113-560-2/02/0011…$5.00.

the design, while section 4 outlines the deployment experience and lessons learned.

2 RELATED WORK

Awareness has been defined as an "understanding of the activities of others, which provides a context for your own activity" [2]). It has been plausibly argued that team coordination and productivity can be enhanced if people can maintain better awareness of the activities of the team and the events in the world that may affect their team (especially with teams that have to work at different times and in different locations [3]).

The literature contains several examples of awareness interfaces. The Sideshow project focuses primarily on peripheral awareness interfaces: a class of awareness interfaces that seeks to provide awareness via software that resides in the user's peripheral attention. Several projects have explored various methods and approaches to providing peripheral awareness.

Awareness on Primary Displays

One area of peripheral awareness research examines methods for embedding peripheral information within a user's primary screen. When utilizing the user's primary monitor, there is often a design tension between wanting to display lots of information and not wanting to take up precious screen real estate. Researchers have addressed this tension in a variety of ways.

Both Elvin [4] and "What's Happening?" [21] are ticker interfaces that reside on the user's main screen and slowly scroll through a variety of pieces of information. Tickers and faders are attractive because they can display lots of information within a small space; however, because they are visually dynamic, there is some debate about how distracting they are. Researchers have studied methods of designing tickers to minimize their level of distraction [12, 14].

Another approach has been to use interfaces that are gently blended into the background of whatever is currently on the screen. The Notification Collage [5] uses this approach to display a variety of information, including shared group documents and status of group members. Harrison et al [7] used a similar approach in a controlled experiment of transparent, layered windows.

Researchers have also explored the use of interfaces that do not guarantee they are always visible. The interfaces are always running on the desktop and are easily accessible, but may be covered by other applications. Georgia Tech's Irwin [13] and Lucent's Rear View Mirror [1] use this approach. Today's popular instant messaging programs (for example, AOL Instant Messenger and Microsoft Windows Messenger) also use this strategy.

Awareness on Secondary Displays

Awareness information can also be displayed on secondary displays. Sometimes the secondary display can be a specialized display like Georgia Tech's InfoCanvas [15], and sometimes the display can be a normal monitor that extends the user's primary workspace to a second display. When studying people who used computers with multiple monitors, Grudin [6] found that often the second displays were not treated as more workspace, but as an area where important information could be displayed peripherally. The Kimura system [10] explores the use of very large secondary displays (in this case, a background wall display) with the purposes of supporting multitasking and awareness.

Ambient Awareness Displays

Ambient awareness displays use the tactic of embedding information into the user's surrounding environment, often without using standard computer screens, and often utilizing the senses of sound and touch (in addition to vision). Perhaps the most famous example of an ambient awareness display is Weiser's twirling string, which kept people aware of network traffic by twisting a small degree each time a packet passed through the network [20]. More recent work includes the Information Percolator, which utilized water tubes and bubbles to display information [8], and the ambientROOM project, which examined various ways of embedding information into artifacts in a typical office environment [9]. The AROMA project [18] also explored a variety of methods to display abstract representations of awareness information, ranging from paintings and landscapes to abstract haptic and kinetic interfaces (electric vibrators, heaters, etc.) to abstract audio (sounds of birds, waterfalls, etc.).

Providing awareness via sound was also the focus the Audio Aura project [16], which explored embedded speech, music, and sound effects. These sound effects provided awareness of e-mail and presence of others and were delivered via portable, wireless headphones. Pacey and MacGregor [17] did a controlled study of using audio and visual cues for conveying awareness information and found that while audio awareness can be especially useful when people are performing a visual task, audio cues can also be more distracting and annoying.

Awareness on Mobile Devices

Much of the literature has focused on providing awareness of information while people are in their office, but with the advent of mobile devices, some researchers have also explored the use of hand-held devices for providing awareness information while people are away from their desks. Tang's Awarenex project [19] examines interfaces and architectures for displaying awareness information on Palm and RIM Blackberry devices. One interesting attribute of some of these devices is that when users are in their offices, they can dock their devices next to their primary display to create a secondary awareness display.

3 SIDESHOW

Of the various methods of providing peripheral awareness, Sideshow provides awareness visually via a sidebar on one's primary display that cannot be covered by other

applications. When a person installs Sideshow, the bar appears on the right side of the Windows desktop (Figure 1a, enlarged image in Figure 2).

The sidebar is filled with a variety of items called "tickets," each of which displays a small summary of information. For example, the ticket pointing to one's Outlook calendar (see top of Figure 2) shows how long one has until the next meeting, as well as the first few words from the meeting title. The ticket pointing to a local camera showing traffic conditions (fourth from the bottom of Figure 2) shows a small, static image from the camera.

The goal of the tickets in the sidebar is to provide a relatively high-level summary of information in a small space. If users decide they want to find out more information about a particular item, they can hover their mouse over a ticket and a large tooltip window appears next to the ticket. One example of a tooltip is shown in Figure 1b (enlarged in Figure 3), which shows what one might see when hovering over a ticket representing a co-worker. The tooltip displays the person's e-mail address, phone number, office number, and calendar for today (if shared). In addition, the person's status for today (whether he has been available, unavailable, or offline, as reported by instant messenger) is shown as a colored band on the right of the calendar.

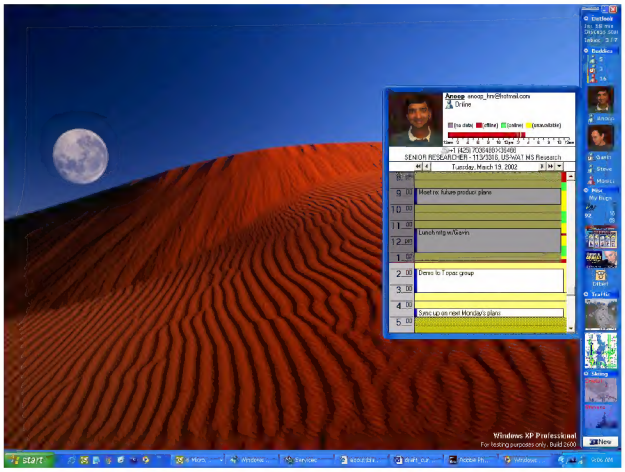
Each of the tooltips can be designed to be highly interactive. For example, in the case of the tooltip for one's e-mail inbox, one can open, reply to, forward, and delete messages from the tooltip. When users are finished interacting with a tooltip, they can make the tooltip disappear by moving their mouse away from the tooltip.

Sideshow tickets also provide an additional method for getting more information: If users double-click a ticket on the sidebar, Sideshow will open an application pointing to a variety of information about the item. For example, double-clicking the Inbox ticket opens Outlook. Double-clicking the "My Bugs" ticket opens the application that teams use to open, discuss, and track issues with the software they're developing.

Managing and Organizing Tickets

Users can add tickets to their sidebar in a variety of ways. One way is to click the "new" button at the bottom of the sidebar, which brings up a wizard interface that allows users to choose and customize a ticket for their use. Users can also drag tickets to their sidebars from e-mail messages, file folders, and web pages. Sideshow is designed such that ticket files can be placed at locations throughout the computing environment so that if users want to watch a piece of information, they know they can drag the associated ticket to their sidebar. For example, Figure 4 shows a mock-up of what an eBay web page might look like with a Sideshow ticket on it. Users could drag the ticket for the auction to their sidebar, and Sideshow would then provide them with updates on the status of the auction.

(a)



(b)

Figure 1: Sideshow is a sidebar on the Windows desktop (a). The bar is filled with items called tickets. Users can get more information about a ticket by hovering their mouse over it, which causes a tooltip to appear. The tooltip for a person ticket is shown in (b). Enlarged images are available in Figures 2 and 3.

As shown in Figure 2, tickets can be organized into groups on the sidebar. For example, the top two tickets are in a group called "Outlook". The next 6 tickets are in the "buddies" group. These groups can be collapsed and expanded. When a group is collapsed, only the title bar of the group is shown, and hovering over the title bar displays a tooltip showing all the tickets inside the group. Users can then hover over the tickets inside the tooltip to get more information about the individual tickets, just as if the group were expanded.

The sizes of the tickets in the sidebar are dynamic. Ticket designers give their tickets a "best size" and a "minimum size." If there is enough room on the sidebar, all tickets are displayed using their "best" sizes. If the sidebar fills up, all the tickets are gradually made smaller until they reach their "minimum" size. If all the tickets reach their "minimum" size, the tickets at the bottom will scroll off the bar.

*I have a meeting in 18 minutes
*There are 3 unread and 7 total messages in my inbox.

* 5 of my buddies are online, 3 are unavailable, and 16 are offline.

* 5 tickets representing individual buddies, some with pictures, some without.

* In my team's bug database, 92 bugs are assigned to me. 1 is high priority, 18 are medium priority, 69 are low priority.

* 5-day forecast for my region

* Current MSNBC headlines

* Today's Dilbert comic strip

* Traffic camera for a local freeway

* Traffic status map for my region

* Two tickets showing current conditions at nearby skiing locations.

Figure 2: Enlarged sidebar from Figure 1.

Buttons to scroll the bar up and down (not shown in the figures) appear when users hover their mouse over the bar.

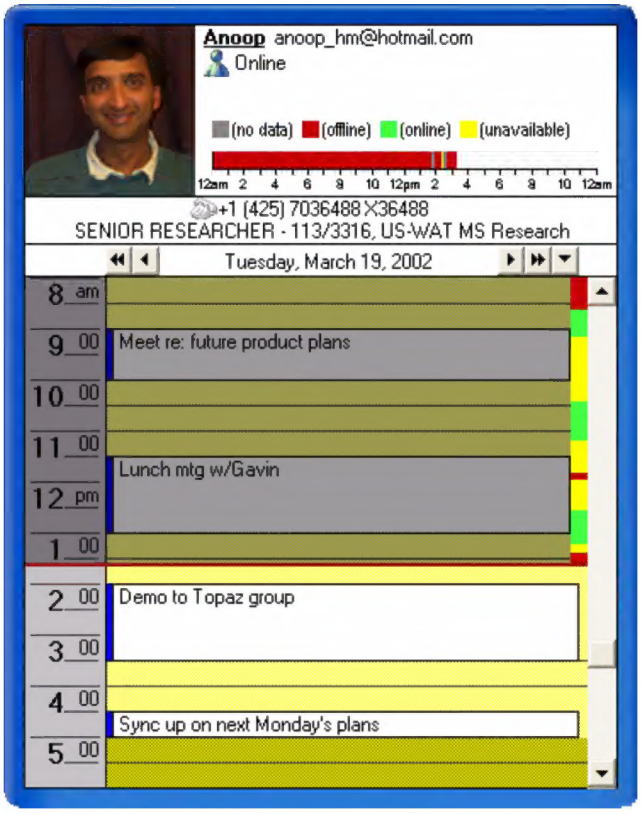


Figure 3: The tooltip for a person ticket.

Users can also change the width of their sidebar to provide Sideshow with more or less space to display tickets. If Sideshow is given enough room, tickets are displayed in multiple columns. When installed, Sideshow is 55 pixels wide—enough room for a single column of tickets.

Tickets that Interrupt

Motion in the visual periphery tends to be distracting, thus Sideshow's tickets are designed to have a minimal amount of motion. This goal is primarily accomplished by changing very few pixels when information is updated. For example, updating the information in several tickets involves changing a few numbers, which affects relatively few pixels. For the more graphical tickets, updates only occur once every few minutes, and if the images need to change, the changes typically are not dramatic.

However, Sideshow users sometimes commented that they wanted Sideshow to distract them (for example, when new e-mail arrived). Thus, Sideshow tickets are also designed with an alert feature. If alerts are turned on, a ticket will display a small window next to it with information about the alert (similar to Microsoft's Instant Messenger notifications that appear when a buddy logs on). For example, when a user receives new e-mail, a small window fades in displaying the sender, subject, and the first few lines of the message. These alert windows can be clicked to get more information (for example, to open the e-mail message that just arrived). The windows can be configured to persist until clicked, or to fade away after a few seconds.

Figure 4: A mock-up of a ticket (circled) on a web page. Tickets can be placed on web pages, and users can drag these tickets to their sidebar to watch different types of information (an eBay auction, in this case).

Alerts were also implemented for the ticket that keeps track of software bugs in a team's bug database. Any time changes are made to bugs of interest, a small window showing the bug appears.

Writing Custom Tickets

Sideshow was designed with the hope that it could be used to watch the majority of the important, dynamic information in one's world. Unfortunately, it did not seem likely that a single software development team would be able to write all the possible tickets people may want to use. In addition, Sideshow faced several of the problems outlined by Maclean et al. in [11]: several of the divisions in the company we studied have custom tools and processes that made writing tickets for each of these divisions extremely difficult. Thus, Sideshow was designed with a model similar to MacLean's: the necessary tools and distribution processes were provided such that one motivated person in a division could write incredibly valuable tickets and then distribute them to her division. Accordingly, a Sideshow SDK (software development kit) was released that allowed people to author tickets using DHTML or C++. In addition, as noted above, tickets were designed such that they could be distributed as files, which enabled people to post tickets on web pages or send them by e-mail.

Authoring a ticket using DHTML was relatively simple: several templates were posted to provide examples of how to obtain data from web pages. In some cases, creating a new ticket was as simple as changing the title of the ticket and the web URL where the ticket got its data.

Design Principles

Several principles were used to inform the design of Sideshow. The first design principle was *make it always present*. Because Sideshow was supposed to utilize peripheral awareness, Sideshow was designed to be an interface that was visually persistent in people's periphery when they were working on their computer. This principle led to the choice of the sidebar as a UI mechanism.

Second, because Sideshow would always be present in the user's visual periphery, Sideshow's second design principle was *minimize motion*. Because of the way people's perceptual systems work, unexpected motion in the visual periphery tends to be highly distracting, thus Sideshow was designed to be as visually calm as possible.

The third design principle was *make it personal*. Sideshow's initial tickets were designed such that the information could be as relevant as possible to people's work (for example, concentrating on e-mail, calendar, and buddies). Furthermore, as noted above, because it is unlikely that a single company or organization could own—or even know about—all the different types of information that people may want to stay aware of, the project's fourth design principle was *make Sideshow extensible*. These principles led to the release of the Ticket SDK and several example tickets early in the project's development.

Sideshow's fifth design principle was *support quick drill-down and escape*. Because of limited space in the user's periphery and the potentially large number of items users may want to stay aware of, it can be difficult for peripheral awareness displays to provide much detail about information the user is watching. However, information might not be useful unless it is detailed, thus Sideshow was designed with the tooltip mechanism so that it would be easy for users to get highly detailed information. The tooltip design was also chosen so that it would be easy to "escape" and return to a prior task after getting more information, in hopes of minimizing the costs of context switching. "Escaping" after drilling down is accomplish by moving the mouse away from the tooltip.

The sixth and last design principle was *make it scalable*. People may want to stay aware of a large number of information sources, thus Sideshow needed to be able to handle dozens of items. This principle led to the creation of groups that could be collapsed, as well as the behavior where tickets first shrink to create more room on the sidebar before resorting to scrolling tickets off the bottom of the bar.

Note that sometimes the principles came in conflict with each other. For example, honoring *make it scalable* meant violating the *make it always present* principle. In this case, the Sideshow team decided it was better to allow people to make their less important tickets less accessible (in a collapsed group or scrolled off the screen) rather than disallowing people from adding tickets once the sidebar filled up, forcing people to make their sidebar bigger, or creating motion by scrolling through sets of tickets.

Sometimes e-mails from the user community also caused the Sideshow team to weigh the value of adhering to design

principles vs. listening to user feedback. For example, the team received numerous requests to give the bar the ability to "auto-hide" (only appear when the mouse touches the edge of the screen). In cases like these, the Sideshow team often implemented the feature but turned it off by default. Analysis of usage logs found that 13% of users enabled the auto-hide feature.

Thus, while the Sideshow team tried to adhere to the principles described in this section, an interesting part of the design process was deciding to break a principle in a particular case and seeing whether it improved the design.

4 DEPLOYING SIDESHOW

Sideshow was first announced January 2001 at our company's annual internal technology fair. The Sideshow team had one of approximately 150 booths at this fair, and during the fair, Sideshow was demonstrated to several hundred attendees. As a result of this fair, about 200 people installed Sideshow.

Figure 5 shows the usage numbers for Sideshow. Nine months after the technology fair, Sideshow had been installed by nearly 4000 people and was being used by nearly 2000 people. After the technology fair in January 2001 and before October 2001, Sideshow was not publicized (aside from maintaining an internal web page and doing private demonstrations to a variety of interested product teams). During these nine months, Sideshow appears to have spread by "word of mouth": co-workers installed Sideshow and told their friends, who installed Sideshow and told their friends, and so on. Sideshow was never demonstrated outside of the United States, thus "word of mouth" was the likely method by which Sideshow spread to employees in over 20 countries throughout North America, South America, Europe, Asia, and Africa.

Figure 5 shows a clear spike in the installation and usage numbers after October. Three events happened at this time. First, on October 23rd, the Sideshow team announced that it was concluding the project and releasing "version 1" (the previous versions had been alpha and beta versions). Second, Sideshow was announced in the company-wide weekly newsletter on November 9th. Third, during the last week of November, the Sideshow team placed posters announcing Sideshow around the company's main campus where approximately 20,000 employees work.

What Contributed to Sideshow's Usage?

It is difficult to say exactly what factors contributed to Sideshow's usage. It is likely that several factors—ranging from Sideshow's design to the iterative development strategy to the company's culture—contributed to such wide use of Sideshow. In this section some data and discussion are provided in an attempt to explain Sideshow's deployment experience.

Two surveys were distributed to Sideshow users in August 2001. The first survey was given to 860 people who were currently using Sideshow; this survey asked about a variety

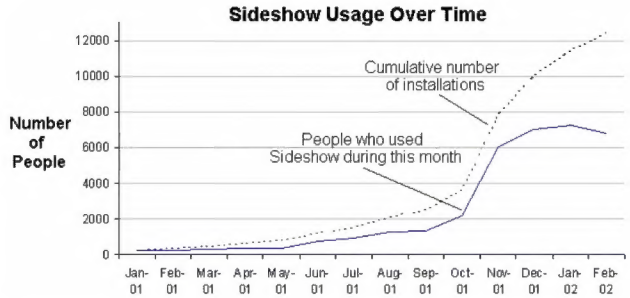


Figure 5: Sideshow usage history. Note that the Sideshow servers were offline for a week in February 2002, thus the numbers for that month may be low. Also note that people could choose to turn off Sideshow's automatic data reporting mechanism, thus actual usage numbers may be higher.

of Sideshow's features. 309 people responded (a 36% response rate). The second survey was given to 698 people who had used Sideshow for more than three days but had not used Sideshow in the prior two weeks. This survey asked a small number of questions about why people stopped using Sideshow. 178 people responded (a 26% response rate). 275 people who downloaded Sideshow but did not use it for at least three days were not surveyed due to concerns about accuracy of self-reported data from these users: many of these people had not seen Sideshow in several months. In retrospect, it would have been useful to automatically send people a survey immediately upon noticing that they had not used Sideshow for the past few weeks (even though this would have created issues of comparing data collected at different times).

Aside from the facts that the survey respondents were from a self-selected population (people who used Sideshow and chose to respond to our survey) and are employees at a large technology company, the survey respondents were relatively diverse. Respondents included administrative assistants, sales staff, finance staff, software developers, product designers, lawyers, and product support professionals; furthermore, respondents were from various locations around the world.

The surveys sought to answer a few major questions. First, if people currently use Sideshow, why do they continue to do so? Second, if they used Sideshow and then stopped, why did they stop? The survey data were also used to examine whether Sideshow's design correctly addressed several of the difficult design tradeoffs.

Data regarding the first question are shown in Figure 6. 26% said they continued to use Sideshow because it made it easy to work with their calendar and e-mail. 20% cited other reasons, which mostly had to do with how Sideshow made it easy for them to stay aware of a variety of information. In particular, users seemed to like how Sideshow allowed them to stay aware of important information without switching away from their primary task. Users wrote:

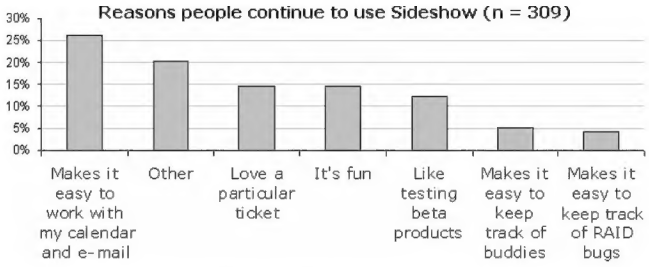


Figure 6: The main reasons users continue to use Sideshow.

> *"I like the quick glance to see amount of mail, bug status, traffic and other info without having to open 10 others apps to get the same info."* – A software developer in the U.S.
>
> *"I love the way it allows me to track lots of information at the same time. I like so many tickets it's impossible to pin one down."* – A sales associate in Canada
>
> *"It's not just one ticket that makes it good, it's the fact that all the important info is presented in the smallest possible space."* – A software tester in the U.S.

Data regarding the question of why people who stopped using Sideshow did so are shown in Figure 7. The most often cited reason for stopping was simply that the prototype was too buggy and not stable enough. These data also provide some explanation for why Sideshow experienced a significant increase in users after the first non-beta version was announced in late October.

In addition to stability issues, survey respondents also listed several "other" reasons for stopping Sideshow use. Many of these reasons had to do with Sideshow's slow startup time (users reported that it would sometimes slow the boot/login process by as much as a minute) and a lack of relevant tickets (especially for users working outside of the company's main corporate campus).

Two issues—screen real estate and distraction—were of particular concern during the Sideshow design process, thus these two issues are worth further scrutiny in the data. 8% of survey respondents said that lost screen real estate was the primary reason they stopped using Sideshow, and 15% cited it as a secondary reason (respondents could mark as many secondary reasons as they desired). When current users of Sideshow were asked whether they thought it was worth giving up the screen space to run Sideshow, respondents' median response was 4 ("agree") out of a 5-point scale (see Table 1). In addition, as mentioned before, although people could to set their sidebars to disappear when the mouse was not over it ("autohide"), only 13% chose to enable this feature.

With regard to Sideshow's level of distraction, 6% of survey respondents cited this as the primary reason they stopped using Sideshow, and 10% cited it as a secondary reason. In addition, current users were asked if they thought Sideshow was distracting. The median score was 2 ("disagree"). Furthermore, when people were asked about what is likely the most distracting part of Sideshow—the alert windows that pop up when new mail arrives—the data remained positive. The median response to the question, "I like being notified by Sideshow as soon as mail arrives" was 4.0 ("agree"). The companion question, "Sideshow's e-mail notifications often distract me from doing important work" received a median score of 2.0 ("disagree"). When asked to comment about the e-mail alerts, users wrote:

> *"That's what I like best about Sideshow. Being able to see who the new mail is from and determine whether I should read it now or not..."* – A usability engineer in the U.S.
>
> *"Love it - being able to do a quick scan to see whether it is an urgent email or not really helps in my role and saves the time previously taken checking Outlook when the new mail icon appears in the tool tray at the bottom of the screen."* – A consultant in the United Kingdom
>
> *"The notification is useful because it prevents me having to go to Outlook as often. Most messages I can read later. Although the popup is distracting, I find its usefulness is worth it."* – A program manager in the U.S.
>
> *"One of the best features in Sideshow"* – A software developer in the U.S.

Thus, although Sideshow was designed with the "minimize motion" principle in mind, and this principle was explicitly violated by the ticket alerting mechanism, it appears that violating the principle was a correct decision to make.

Ticket Feedback & Authoring

The usefulness of Sideshow is likely tied to the usefulness of available tickets, thus current Sideshow users were also asked about the tickets they used. Table 2 shows the ratings of ticket usefulness by current users of Sideshow. While some tickets were rated higher than others, it appears as if no single ticket was overwhelmingly the most popular

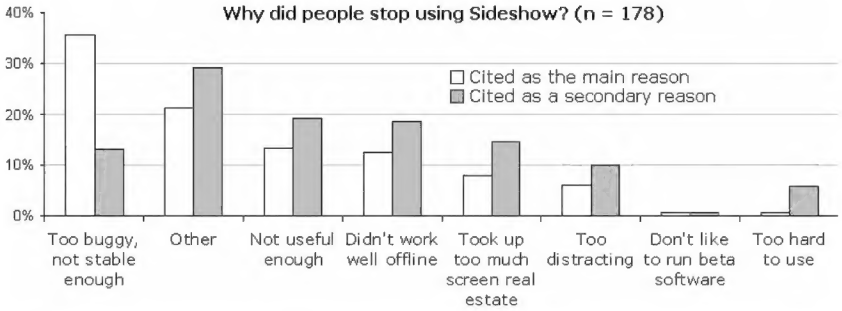


Figure 7: The reasons people said they stopped using Sideshow. People could select one primary reason and as many secondary reasons as they desired.

Table 1: Selected questions from the survey of people currently using Sideshow (n = 309). Responses:

1 = Strongly Disagree
2 = Disagree
3 = Neither agree nor disagree
4 = Agree
5 = Strongly Agree

Question	Median	Avg	Std Dev
Sideshow is distracting	2.0	2.3	0.9
Sideshow interrupts me when I'm trying to do other work.	2.0	2.2	0.9
It's worth giving up the screen space to run Sideshow.	4.0	3.8	0.9
Sideshow grabs my attention at the right times.	4.0	3.6	0.8
Sideshow helps me stay aware of information that's critical for me to keep track of.	4.0	3.7	0.8
I like being notified by Sideshow as soon as new mail arrives.	4.0	4.1	1.0
Sideshow's e-mail notifications often distract me from doing important work.	2.0	2.3	0.9

ticket. Although the data in Figure 6 show that many people primarily continued to use Sideshow because of the Outlook folder and calendar tickets, the data in Table 2 show that other tickets were just as popular. In addition, as noted earlier, many people who marked "other" in Figure 6 did so because they continued to use Sideshow because it helped them stay aware of a variety of information—a notion further supported in Table 2 by the high ratings of several tickets. Yet another interesting note from Table 2 are the similarly high ratings for tickets that helped people stay aware of business/productivity information (e-mail, calendar, software bugs) and personal/non-business information (weather, traffic).

Note that Table 2 only displays data for tickets created by the Sideshow team. As noted in section 3, customized tickets could also be authored using C++ or DHTML. Currently, Sideshow's ticket library has approximately 100 tickets for 17 different regions (Australia, Germany, San Francisco, Los Angeles, Michigan, North Dakota, etc.). 83 of these tickets were written by people not on the Sideshow team, and of these 83 tickets, only three are known to have been created using the C++ SDK; the remainder were authored using DHTML templates provided by Sideshow. In addition, Sideshow's library only contains tickets that were sent to the Sideshow team; it is possible that additional tickets were created and passed around to teams via e-mail or web pages.

Some examples of tickets created by others include status of baseball games, news from various sources (newspapers in

Table 2: Ratings of ticket usefulness by current Sideshow users. Data are only reported for survey respondents who tried the tickets in question (thus the varying N numbers). Response and their codes were:
1 = Tried, but did not like this ticket
2 = Like this ticket a little
3 = Like this ticket a lot
4 = Love this ticket

Ticket	N	Median	Avg	Std Dev
Regional traffic status map	192	3.0	3.1	0.9
Outlook calendar	288	3.0	3.0	0.9
Stock market	290	3.0	3.0	0.8
Outlook mail folder	291	3.0	2.9	0.9
Current weather conditions	250	3.0	2.9	0.9
Traffic camera	199	3.0	2.9	0.9
Software bugs	112	3.0	2.8	0.9
Instant Messenger Buddies	280	2.5	2.6	0.9
Person	190	3.0	2.6	0.9
5-day forecast	142	3.0	2.6	1.0
Cafeteria menu	52	2.0	2.4	0.9
Current Doppler radar (weather)	126	2.0	2.4	0.9
eBay auction	27	2.0	2.3	1.0
MSNBC Headlines	211	2.0	2.2	0.9
File folder changes	54	2.0	2.0	1.0

Argentina, Australia, India, Portugal, etc.), traffic status for various regions, assorted daily comic strips (Dilbert, Snoopy, Garfield, etc.), and conditions at nearby ski resorts. Other tickets were created to facilitate collaborative group work. For example, the product support organization created a set of tickets to help them track open issues, phone queues, and other team information. Another team created a ticket to help them track the task of running several dozen computers through a daily set of test cases for the latest build of their team's software.

Other Factors Contributing to Success

While Sideshow's design was likely the most important success factor, looking back, several of the deployment strategies the team used also seemed to contribute to a successful deployment. Specifically, an iterative design approach is often recognized as a critical component of making useful, usable software, and the Sideshow team took several steps to try to create an iterative design cycle during Sideshow's deployment.

First, the Sideshow team tried to release new versions of Sideshow frequently. These releases contained a variety of bug fixes, new tickets, and new features, and were often guided by feedback from the user community. During the nine month period when Sideshow was being actively

developed, 22 releases were made (an average of one release every 12.4 calendar days). These rapid releases provided the team with the ability to quickly try new features and gauge their success via usage logs and e-mail feedback. For example, the Sideshow team was divided over whether tickets should be provided with the alerting feature (the feature where tickets could pop up a window to notify users that something of interest had happened). The Sideshow team felt that such a feature would violate the "minimize motion" principle and strayed from the focus on peripheral awareness. However, after the feature was deployed (and turned on by default for the Outlook Inbox ticket), no negative e-mails were received about the feature. In contrast, other newly added features (like the first implementation of ticket groups) would sometimes generate several negative messages from the user community.

Another example of a feature added hesitantly by the Sideshow team was the ability for Sideshow to work in "offline" mode (when the computer is not connected to the network). Because the point of Sideshow is to display live data, the team felt that creating an "offline" mode where Sideshow would display static data would be pointless. However, laptop users indicated that they often wanted to see the last known status of their information (even if the data were stale), and that they did not want to have to stop and restart Sideshow every time they disconnected from and connected to the network. Thus, the feature was added.

Second, the Sideshow team tried to keep a constant dialog with its user community. Each time a new release was made available, it was announced via a "Sideshow announcements" e-mail distribution list and feedback was encouraged. In addition, a "Sideshow feedback" e-mail distribution list was created, and links to this distribution list were placed in the Sideshow application and on Sideshow's internal web page. Users could also submit bug reports directly to Sideshow's bug database. As a result, during the nine month Sideshow development period, the Sideshow team received approximately 895 feedback e-mail messages (an average of 4.6 messages per work day). The Sideshow team also tried to provide excellent customer service, thus approximately 75% of these messages received responses (within one day if possible). Eventually the amount of e-mail received became so overwhelming that an automated, web-based feedback system was created.

Third, the Sideshow application was designed to be very easy to update. The Sideshow application periodically checks a server for a new version, and when one is detected, it asks the user if he would like to update. If he agrees, the application closes, updates, and restarts. If he defers the update, Sideshow updates itself at night (presumably when most people are not at their desks).

These three strategies—frequent updates, a constant dialog with the user community, and easy updating—seemed to create the iterative design loop that contributed to Sideshow's design and its success. In fact, it might have created a sense of ownership among the user community (people would often make suggestions and see them implemented in a few weeks), which might have led to people telling their friends to try Sideshow. People could also participate in the Sideshow development effort by writing tickets and having them published in the Sideshow library, which may have led to a similar sense of ownership.

As a software development organization, the company's culture also likely contributed to Sideshow's iterative design effort. The company has a strong culture where people are encouraged, expected, and sometimes forced to try beta versions of upcoming products in an effort to help improve the design and find all the bugs. This culture likely helped create an environment where a research project like Sideshow could flourish.

5 CONCLUDING REMARKS

In this paper we described a peripheral awareness system that can deliver awareness and notifications of the status of people, objects shared by group members, and information of almost any kind appearing on the Web, delivered via the Internet, or accessible on an intranet. The very rapid, bottom-up adoption by thousands of employees willing to permanently devote display real estate to Sideshow is an early demonstration of the importance of this class of information, something long anticipated by the CSCW community but not demonstrated on this scale.

We described the design principles employed to identify and enable the key features—iterative deployment-driven design—and the refined feature set that is currently in use. We have outlined the concrete design tradeoffs involved in specific features and shown the combination of principles and features that provided value to end users.

Sideshow is in continual use by thousands of employees. Potential future research directions include studies involving uses of Sideshow in other domains (for example, for home users or users at non-software companies) and studies of Sideshow on mobile devices. Additional research directions involve methods to help users customize Sideshow without a lot of effort. For example, if Sideshow notices that a person visits a document that is often edited by other people, it could place a ticket to watch the document in a "recommended tickets" group on the sidebar. Similarly, if a ticket's information has not changed in quite a while, it could suggest that the ticket be removed.

Questions remain about how people use Sideshow, how they customize it, and whether it affects work practices for teams and individuals. Now that Sideshow is more stable, with more users and more tickets written by the user community, follow-up surveys are appropriate. Usage data collected by the Sideshow system can be analyzed in detail, and entire teams using Sideshow can be interviewed. Prototype and commercial systems with these features are likely to proliferate, providing a need for deeper research into the costs and benefits of awareness and notification.

ACKNOWLEDGMENTS

We thank Michael Boyle for writing the first prototype of Sideshow as an intern in our group during Summer 2000.

REFERENCES

1. Atkins, D., Boyer, D., Handel, M., Herbsleb, J., Mockus, A., Wills, G. The Product Development Collaboratory at Lucent Technologies. http://www.bell-labs.com/org/11359/colab_prod/
2. Dourish, P., and Bellotti, V. (1992). Awareness and Coordination in Shared Workspaces. *Proceedings of the ACM Conference on Computer Supported Cooperative Work* (CSCW 1992), 107-114.
3. Dourish, P., and Bly, S. (1992). Portholes: Supporting Awareness in a Distributed Work Group. *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI 1992), 541-547.
4. Fitzpatrick, G., Kaplan, S., Mansfield, T., Arnold, D., and Segall, B. (2000). Supporting Public Availability and Accessibility with Elvin: Experiences and Reflections. *Computer Supported Cooperative Work: The Journal of Collaborative Computing*.
5. Greenberg, S., and Rounding, M. (2001). The Notification Collage: Posting Information to Public and Personal Displays. *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI 2001), 514-521.
6. Grudin, J. (2001). Partitioning Digital Worlds: Focal and Peripheral Awareness in Multiple Monitor Use. *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI 2001), 458-465.
7. Harrison, B., Ishii, H., Vicente, K., and Buxton, W. (1995). Transparent Layered User Interfaces: An Evaluation of a Display Design to Enhance Focused and Divided Attention. *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI 1995), 317-324.
8. Heiner, J., Hudson, S., and Tanaka, K. (1999). The Information Percolator: Ambient Information Display in a Decorative Object. *Proceedings of the ACM Symposium on User Interface Software and Technology* (UIST 1999), 141-148.
9. Ishii, H., and Ullmer, B. (1997). Tangible Bits: Towards Seamless Interfaces between People, Bits, and Atoms. *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI 1997), 234-241.
10. MacIntyre, B., Mynatt, E., Voida, S., Hansen, K., Tullio, J., and Corso, G. (2001). Support for Multitasking and Background Awareness Using Interactive Peripheral Displays. *Proceedings of the Symposium on User Interface Software and Technology* (UIST 2001), 41-50.
11. MacLean, A., Carter, K., Lovstrand, L, and Moran, T. (1990). User-Tailorable Systems: Pressing the Issues with Buttons. *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI 1990), 175-182.
12. McCrickard, D., Catrambone, R., and Stasko, J. (2001). Evaluating Animation in the Periphery as a Mechanism for Maintaining Awareness. *Proceedings of the IFIP TC.13 Conference on Human Computer Interaction* (Interact 2001), 148-156.
13. McCrickard, D. (1999). Maintaining Information Awareness with Irwin. *Proceedings of the World Conference on Educational Multimedia/Hypermedia and Educational Telecommunications* (ED-MEDIA 1999).
14. Maglio, P., and Campbell, C. (2000). Tradeoffs in Displaying Peripheral Information. *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI 2000), 241-248.
15. Miller, T., and Stasko, J. (2001). The InfoCanvas: Information Conveyance through Personalized, Expressive Art. *Extended Abstracts from the ACM Conference on Human Factors in Computing Systems* (CHI 2001), 305-306.
16. Mynatt, E., Back, M., Want, R., Baer, M., and Ellis, J. (1998). Designing Audio Aura. *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI 98), 566-573.
17. Pacey, M., and MacGregor, C. (2001). Auditory Cues for Monitoring a Background Process: A Comparative Evaluation. *Proceedings of the IFIP TC.13 Conference on Human Computer Interaction* (Interact 2001).
18. Pedersen, E., and Sokoler, T. (1997). AROMA: Abstract Representation of Presence Supporting Mutual Awareness. *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI 1997), 51-58.
19. Tang, J., Yankelovich, N., Begole, J., Van Kleek, M., Li, F., and Bhalodia, J. (2001). ConNexus to Awarenex: Extending Awareness to Mobile Users. *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI 2001), 221-228.
20. Weiser, M., and Brown, J. (1996). Designing Calm Technology. *PowerGrid Journal*, v1.01, July 1996. (see http://nano.xerox.com/hypertext/weiser/calmtech/calmtech.htm
21. Zhao, Q. (2001). Opportunistic Interfaces for Promoting Community Awareness. Georgia Institute of Technology PhD thesis.